DEEP LEARNING NEURAL NETWORK STRUCTURE OPTIMIZATION*

M. S. Potanin[1], K. O. Vayser[2], V. A. Zholobov[3], V. V. Strijov[4]

Abstract: The paper investigates optimal model structure selection problem. The model is a superposition of generalized linear models. Its elements are linear regression, logistic regression, principal components analysis, autoencoder and neural network. Model structure refers to values of structural parameters that determine the form of final superposition. This paper analyzes model structure selection method and investigates dependence of accuracy, complexity and stability of model on it. The paper proposes an algorithm for selection of neural network optimal structure. The proposed method was tested on real and synthetic data. Experiment results

*This research was supported by RFBR, projects проекты 19-07-1155, 19-07-0885, and by Government of the Russian Federation, agreement 05.Y09.21.0018. This paper contains results of the project Statistical methods of machine learning, which is carried out within the framework of the Program "Center of Big Data Storage and Analysis" of the National Technology Initiative Competence Center. It is supported by the Ministry of Science and Higher Education of the Russian Federation according to the agreement between the M. V. Lomonosov Moscow State University and the Foundation of project support of the National Technology Initiative from 11.12.2018, No 13/1251/2018.

[1]Moscow Institute of Physics and Technology, mark.potanin@phystech.edu

[2]Moscow Institute of Physics and Technology, vajser.ko@phystech.edu

[3]Moscow Institute of Physics and Technology, zholobov.va@phystech.edu

[4]A. A. Dorodnicyn Computing Center, Federal Research Center «Computer Science and Control» of the Russian Academy of Sciences, Moscow Institute of Physics and Technology, strijov@ccas.ru

in significant structural complexity reduction of model while maintaining the accuracy of approximation.

Keywords: model selection; linear models; autoencoders; neural networks; structure; genetic alghorithm

References

[1] Bakhteev, O. Yu, and V. V. Strijov. 2018. Deep learning model selection of suboptimal complexity. *Automation and Remote Control*. 79.8:1474–1488.

[2] Katrutsa, A. M., and V. V. Strijov. 2015. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*. 142:172–183.

[3] LeCun, Y., J. S. Denker, and S. A. Solla. 1990. Optimal brain damage. *Advances in neural information processing systems 2*.

[4] Rao, C. R., *et al*. 1973. Linear statistical inference and its applications. Vol. 2. New York: Wiley.

[5] Glorot, X., and Yo. Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*.

[6] Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*. 2.4:303–314.

[7] Frank, A., and A. Asuncion. 2010. UCI Machine Learning Repository, Available at: http://archive.ics.uci.edu/ml

[8] Dong, X., S. Chen, and S. Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*.

[9] Hassibi, B., and D. G. Stork. 1993. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*.

[10] Molchanov, P., *et al.* 2016. Pruning convolutional neural networks for resource efficient transfer learning. arXiv preprint arXiv:1611.06440 3.

[11] Chaber, P., and M. Ławryńczuk. 2018. Pruning of recurrent neural models: an optimal brain damage approach. Nonlinear Dynamics 92.2:763–780.

[12] Han, S., H. Mao, and W. J. Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

[13] Kingma, D. P., T. Salimans, and M. Welling. 2015. Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems*.

[14] Molchanov, D., A. Ashukha, and D. Vetrov. 2017. Variational dropout sparsifies deep neural networks. *Proceedings of the 34th International Conference on Machine Learning*. PMLR. 70:2498–2507.

[15] Arabasadi, Z., *et al.* 2017. Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. *Computer methods and programs in biomedicine*. 141:19–26.

[16] Hinton, G. E., and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*. 313.5786:504–507.

[17] Srivastava, N., *et al.* 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. 15.1:1929–1958.

[18] Elsken, T., J. H. Metzen, and F. Hutter. 2018. Neural architecture search: A survey. arXiv preprint arXiv:1808.05377.

[19] Hutter, F., L. Kotthoff, and J. Vanschoren. 2019. Automated Machine Learning-Methods, Systems, Challenges. Automated Machine Learning.

[20] Potanin, M. S., K. O. Vajser, V. A. Zholobov, and V. V. Strijov. 2019. Appendix: computational expirement and basic theorems. Available at: https://github.com/MarkPotanin/GeneticOpt

Оптимизация структуры сетей глубокого обучения*

М. С. Потанин[1], **К. О. Вайсер**[2], **В. А. Жолобов**[3], **В. В. Стрижов**[4]

Аннотация

Исследуется проблема выбора оптимальной структуры модели. Моделью является суперпозиция обобщенных линейных моделей, элементами которой являются линейная регрессия, логистическая регрессия, метод главных компонент, автоэнкодер и нейросеть. Под структурой модели понимаются значения структурных параметров модели, задающих вид итоговой суперпозиции. Исследуются свойства алгоритма выбора структуры модели. Исследуются зависимость точности, сложности и устойчивости модели от способа задания структуры. Создан алгоритм выбора оптимальной структуры нейронной сети. Проведен вычислительный эксперимент с использованием реальных и синтетических данных. В результате эксперимента существенно понижена структурная сложность моделей с сохранением точности аппроксимации.

Ключевые слова: выбор моделей; линейные модели; автокодировщик; нейронные сети; структура; генетический алгоритм

*Работа выполнена при поддержке РФФИ (проекты 19-07-1155, 19-07-0885) и правительства РФ (соглашение 05.Y09.21.0018). Настоящая статья содержит результаты проекта «Статистические методы машинного обучения», выполняемого В рамках реализации Программы Центра компетенций Национальной технологической инициативы «Центр хранения и анализа больших данных», поддерживаемого Министерством науки и высшего образования Российской Федерации по Договору МГУ им. М. В. Ломоносова с Фондом поддержки проектов Национальной технологической инициативы от 11.12.2018 № 13/1251/2018.

[1]Московский физико-технический институт, mark.potanin@phystech.edu

[2]Московский физико-технический институт, vajser.ko@phystech.edu

[3]Московский физико-технический институт, zholobov.va@phystech.edu

[4]Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук; Московский физико-технический институт, strijov@ccas.ru

Введение

Решается задача аппроксимации выборки нейронными сетями. Задана выборка, множество пар $(\mathbf{x}, y)$, $\mathbf{x} \in \mathbb{X} = \mathbb{R}^n$, $y \in \mathbb{Y}$, где y — зависимая переменная, а $\mathbf{x}$ — признаковое описание объекта. В случае задачи классификации $\mathbb{Y} = \{1, \ldots, M\}$. В случае задачи регрессии $\mathbb{Y} \subseteq \mathbb{R}$. Нейронная сеть является универсальной моделью [6], так как приближает произвольную непрерывную функцию многих переменных с любой точностью. Нейрон, или однослойная нейронная сеть, являются суперпозицией двух функций — функции активации и линейной комбинации признаков объекта. Но однослойные сети применимы только для линейно разделимых выборок. Для апроксимации выборок общего вида требуется универсальная модель, оптимизация структуры которой и исследуется в данной работе.

Теорема 1 (Колмогоров, 1961) в [20] утверждает, что функцию от n аргументов предствима в виде комбинации $n(2n + 1)$ функций одного аргумента. Какими именно должны быть функции σ_i, g_{ij}, не указывается. Теорема об универсальной аппроксимации 2 (Цыбенко, 1989) в [20] утверждает, что искусственная нейронная сеть прямой связи, в которой связи не образуют циклов, с одним скрытым слоем аппроксимирует любую непрерывную функцию многих переменных с любой точностью. Однако затруднительно выбрать такую структуру нейронной сети, чтобы размеры скрытого слоя не были велики. В теореме 3 (Ханин, 2017) оценивается оптимальная размерность скрытых слоев и обосновывается возможность замены нейронной сети с функциями активации ReLU [20] с входным слоем размерности n и одним скрытым слоем размерности k на эквивалентную с глубиной $k+2$ и размерностями скрытых слоев $n+2$. Эти три теоремы и определяют исследуемую структуру суперпозиций сети глубокого обучения.

Исследуется зависимость ошибки от суперпозиции автокодировщиков [16] и многослойной нейронной сети. Ошибка состоит из двух слагаемых: ошибки восстановления элементов выборки после кодирования и восстановления зависимых переменных. Слагаемые используют одни и те же признаки объектов, которые являются независимыми переменными, но разные зависимые переменные. Для автокодировщика зависимая переменная — это сами признаки объекта, для нейронной сети, следующей за ним, зависимая переменная — ответ y на объекте. Точка разделения — это место в суперпозиции, где автокодировщик, имеющий оптимальные параметры, передает преобразованный вектор признаков в нейросеть. Необходимо найти оптимальное расположение разделения автокодировщика и сети, которое минимизирует ошибку аппроксимации выборки. Под структурой такой модели понимаются величины, задающие вид итоговой суперпозиции — то есть число слоев автокодировщика и нейросети, а также число нейронов в слоях.

Процедура минимизации ошибки аппроксимации выборки следующая: сначала максимизируется точность реконструкции кодировщиков, затем оптимизируются параметры нейросети.

Вместе с точностью оптимизируется сложность модели. Под сложностью понимается *структурная сложность модели* — это число параметров модели, с учетом их области определения. Альтернативой этому определению является *статистическая*

сложность модели — минимальная длина описания, т.е. минимальное количество информации, которое требуется для передачи информации о модели и о выборке [1].

В данном исследовании для выбора оптимальной структуры используется генетический алгоритм. Задается множество случайных начальных значений структурных параметров. Затем вычисляется значение функции ошибки аппроксимации, которое характеризует качество модели в наборе. Согласно этой функции выбираются модели, которые обмениваются структурными параметрами, образуя новую структуру. Многократное повторение этой операции позволяет получить оптимальную структуру модели. В [15] для распознавания сердечно-сосудистых заболеваний используются генетический алгоритм и нейронная сеть для отбора признаков и оптимальной настройки матрицы параметров.

Нейронные сети требуют больших вычислительных ресурсов и памяти, что затрудняет их развертывание во встроенных системах с ограниченными аппаратными ресурсами [12]. Для снижения сложности нейронных сетей используется идея редукции лишних элементов. Например, прореживают ветви решающих деревьев или параметры в нейронной сети. Основная идея метода — находится подмножество параметров, изменение которых не влияет на значение функции ошибки. Тогда эти параметры обнуляются и используются оставшиеся параметры, то есть это способ перейти от полносвязной сети к неполносвязной. Прореживание сети используется для повышения обобщающей способности.

Выяснить без полного перебора, какие параметры важны для предсказания, а какие — нет затруднительно. Для этого используются алгоритмы регуляризации, например — OBD [3] и OBS [9]. Они используют производные второго порядка функции ошибки по параметрам для выбора удаляемых параметров и оценки их «значимости». Дополнительный способ применения этого метода — это отбор признаков. Можно посчитать значимость не только для отдельного параметра, но и для входного признака. Затем упорядочив признаки по величинам их значимости, можно обнулить все параметры, которые используют значения этого признака, и тем самым удалить признак из модели.

В [8] авторы предлагают новый метод прореживания для глубоких нейронных сетей. Параметры каждого слоя независимо прореживаются на основе производных второго порядка функции послойной ошибки по соответствующим параметрам. В работе [10] используются производные первого порядка для снижения сложности сверточных нейронных сетей. Использование [3] позволило в [11] уменьшить число структурных параметров рекуррентной нейронной сети на 60% и снизить ошибку на валидационной выборке на 30%, по сравнению с исходной моделью.

Существуют автоматизированные методы поиска нейросетевой архитектуры [18], которые являются частью парадигмы автоматического машинного обучения [19]. Система поиска получает на вход набор данных и тип решаемой задачи, и результатом является архитектура нейронной сети, которая будет работать лучше всех других архитектур для данной задачи при обучении на предоставленном наборе данных.

В [12] представлен подход глубокого сжатия, состоящий из трех этапов. Уменьше-

ние числа параметров, путем исключения нулевых параметров. Затем нейронная сеть квантизируется. Последним этапом является применение кодирования Хаффмана. После первых двух шагов сеть обучается заново, для более точной настройки параметров. В наборе данных ImageNet данный метод позволяет уменьшить объем памяти, требуемый сетью AlexNet, в 35 раз, с 240 Mb до 6,9 Mb, без потери точности. Размер VGG-16 уменьшается в 49 раз с 552 Mb до 11,3 Mb, опять же без потери точности. Это позволяет использовать сложные нейронные сети на мобильных устройствах.

В статье [13] было предложено использовать вариационное исключение [17] для настройки величины исключения отдельно для каждого нейрона. Применение этого метода [14] позволяет уменьшить количество параметров в 280 раз на архитектуре нейросети LeNet и в 68 раз в VGG-подобных нейросетях без значительного снижения качества аппроксимации.

Работа состоит из постановки решаемой задачи, описания задачи выбора оптимальной структуры модели, за которыми следует определение функции ошибки и критериев качества модели. Далее следует вычислительный эксперимент, содержащий функциональную схему программной системы, описание наборов данных, используемых подходов решения задачи и результаты работы предложенного метода.

Постановка задачи выбора модели

Задана выборка

$$(\mathbf{x}_i, y_i), \quad \mathbf{x}_i \in \mathbb{R}^n, \quad y_i \in \mathbb{R}^1, \quad i = 1, \ldots, m, \tag{1}$$

где $\mathbf{x}$ — описание объекта, вектор из d элементов признаков, y — зависимая переменная. Моделью называется отображение $f : (\mathbf{x}, \mathbf{w}) \mapsto y$. Требуется построить аппроксимирующую модель $f(\mathbf{x})$ вида:

$$f = \sigma_k \circ \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}} \boldsymbol{\sigma}_{k-1} \circ \mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \circ \underset{n_2\times 1}{\mathbf{W}_2}\boldsymbol{\sigma}_1 \circ \underset{n_1\times n}{\mathbf{W}_1}\underset{n\times 1}{\mathbf{x}}. \tag{2}$$

Эта модель рассматривается как суперпозиция линейной модели, глубокой нейросети и автоэкнодера. Рассмотрим различные модели как частные случаи (2).
Линейная или логистическая регрессия и один нейрон — имеют вид

$$f(\mathbf{x}, \mathbf{w}) = \boldsymbol{\sigma}(\mathbf{w}^{\mathsf{T}}\mathbf{x}), \tag{3}$$

где $\boldsymbol{\sigma}$ — функция активации, непрерывная монотонная дифференцируемая функция (5), $\mathbf{w}$ — вектор параметров, $\mathbf{x}$ — объект, вектор с присоединенным элементом единица соответствующим аддитивному параметру w_0. При использовании линейной функции активации, получаем линейную регрессию

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}. \tag{4}$$

Такую функцию активации мы обозначим $\boldsymbol{\sigma} = \mathbf{id}$. При использовании сигмоидной функции активации, получаем модель логистической регрессии

$$f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x})}. \tag{5}$$

Двухслойная нейронная сеть, состоящая из линейной комбинации нейронов, однослойных нейронных сетей

$$f(\mathbf{x},\mathbf{w}) = \sigma^{(2)}\left(\sum_{i=1}^{n_2} w_i^{(2)} \cdot \sigma^{(1)}\left(\sum_{j=1}^{n} w_{ij}^{(1)} x_j + w_{i0}^{(1)}\right) + w_0^{(2)}\right) = \sigma \circ \mathbf{w}^{\mathsf{T}}\boldsymbol{\sigma} \circ \mathbf{W}\mathbf{x}. \tag{6}$$

Метод главных компонент. Модель допускает вращения признакового пространства, то есть объекты преобразовываются только с помощью поворотов:

$$\mathbf{h} = \mathbf{W}\mathbf{x}, \tag{7}$$

где $\mathbf{W}$ — матрица поворота. Она ортогональна:

$$\mathbf{W}\mathbf{W}^{\mathsf{T}} = \mathbf{I}_n. \tag{8}$$

Полученное пространство образов $\mathbf{h}$ называется скрытым. Происходит преобразование без потерь.

При удалении нескольких строк оптимальной [4] матрицы $\mathbf{W}$, например их число $u < n$, полученный вектор $\mathbf{h}$ имеет размер $u \times 1$. Получается проекция $\mathbf{h}$ вектора $\mathbf{x}$. Согласно теореме Рао С.Р. [4], первые u главных компонент восстанавливают $\mathbf{h}$ оптимальным способом,

$$\mathbf{r}(\mathbf{x}) = \mathbf{W}^{\mathsf{T}}\mathbf{h}. \tag{9}$$

Автокодировщик $\mathbf{h}$ — это монотонное нелинейное отображение входного вектора свободных переменных $\mathbf{x} \in \mathbb{R}^n$ в скрытое представление $\mathbf{h} \in \mathbb{R}^u$ вида:

$$\mathbf{h}(\mathbf{x}) = \boldsymbol{\sigma}(\underset{u\times n}{\mathbf{W}}\mathbf{x} + \mathbf{b}). \tag{10}$$

В случае $\boldsymbol{\sigma} = \mathbf{id}$ и (8) автокодировщик тождественен методу главных компонент. Скрытое представление $\mathbf{h}$ реконструирует вектор $\mathbf{x}$ линейно:

$$\mathbf{r}(\mathbf{x}) = \underset{n\times u}{\mathbf{W}'}\mathbf{h} + \mathbf{w}_0'. \tag{11}$$

Задача выбора оптимальной структуры модели

Решается задача выбора оптимальной структуры модели

$$f = \sigma_k \circ \boldsymbol{\Gamma}_k \otimes \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}}\boldsymbol{\sigma}_{k-1} \circ \boldsymbol{\Gamma}_{k-1} \otimes \mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \circ \boldsymbol{\Gamma}_2 \otimes \underset{n_2\times 1}{\mathbf{W}_2}\boldsymbol{\sigma}_1 \circ \boldsymbol{\Gamma}_1 \otimes \underset{n_1\times n}{\mathbf{W}_1}\underset{n\times 1}{\mathbf{x}}, \tag{12}$$

где $\boldsymbol{\Gamma}$ — матрица, задающая структуру модели; $\otimes$ — адамарово произведение, определяющееся как поэлементное умножение. Если элемент $\gamma \in \{0,1\}$ матрицы $\boldsymbol{\Gamma}$ равен нулю, то соответствующий элемент матрицы параметров $\mathbf{W}$ обнуляется, и не участвует в работе модели. Множество индексов соответствующих ненулевым элементам матрицы $\boldsymbol{\Gamma}$

обозначается $\mathcal{A}$. Требуется найти такое подмножество индексов $\mathcal{A}^*$, которое доставляет минимум функции:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}\subseteq\mathcal{I}} S(f_{\mathcal{A}}|\mathbf{w}^*, \mathfrak{D}_{\mathcal{C}}), \tag{13}$$

на разбиении выборки $\mathfrak{D}$, определенным множеством индексов $\mathcal{C}$. Здесь $\mathcal{I} = \mathcal{C} \sqcup \mathcal{L}$ — все индексы всех матриц $\mathbf{\Gamma}$. То есть требуется снизить число признаков и повысить устойчивость модели.

При этом параметры $\mathbf{w}^*$ модели доставляют минимум ошибки:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} S(\mathbf{w}|\mathfrak{D}_{\mathcal{L}}, f_{\mathcal{A}}), \tag{14}$$

на разбиении выборки, определенной множеством $\mathcal{L}$. Процедура разбиения описана в вычислительном эксперименте.

Генетический алгоритм. Для решения задачи оптимизации структуры (13) используется генетический алгоритм. Структура нейронной сети (12) включает в себя k слоев, l-ый слой содержит N_l нейронов, $\sum_{l=1}^{k} N_l = L$. Каждому слою соответствует матрица $\mathbf{\Gamma}_l \in \{0,1\}^{N_l}$. Это означает, что параметры, которые умножаются поэлементно на ноль, не будут учитываться. Составляется вектор

$$\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_L] = \mathbf{vec}[\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \ldots, \mathbf{\Gamma}_k],$$

соответствующий (12). Процедура оптимизации струкруты:

1. Задается множество начальных значений $\mathfrak{G} = \{\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \ldots \boldsymbol{\gamma}_R\}$, где случайным образом задаются элементы вектора бинарного вектора $\boldsymbol{\gamma}$.
2. Для каждого $\boldsymbol{\gamma}_\iota \in \mathfrak{G}$ вычисляется значение функции ошибки S (16).
3. Для каждого $\boldsymbol{\gamma}_\iota$ оценивается вероятность выбора его как структуры для скрещивания с помощью функции: $P_\iota = \frac{1/S_\iota}{\sum_{\iota=1} 1/S_\iota}$. Выбирается пара структур $\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_q$ с максимальной вероятностью.
4. Выбирается случайный индекс точки разделения $\nu \in \{1, ..., L-1\}$.
5. Структуры разделяются на две части, происходит обмен элементами, следующими за ν:
$$[\gamma_{p,1}, \ldots, \gamma_{p,\nu}, \gamma_{q,\nu+1}, \ldots, \gamma_{q,L}] \to \boldsymbol{\gamma}_p^{'},$$
$$[\gamma_{q,1}, \ldots, \gamma_{q,\nu}, \gamma_{p,\nu+1}, \ldots, \gamma_{p,L}] \to \boldsymbol{\gamma}_q^{'}.$$
6. Выбираются случайные номера $\eta_1, \ldots, \eta_Q \in \{1, \ldots, L\}$.
7. У векторов $\boldsymbol{\gamma}_p^{'}, \boldsymbol{\gamma}_q^{'}$ инвертируются позиции с номерами $\eta_1, \ldots, \eta_Q$.
8. Пункты 4–8 повторяются $R/2$ раз. Множество $\mathfrak{G}$ содержит на каждой итерации R структур, которым соответствует наименьшее ошибка.

Здесь R, Q — фиксированные параметры алгоритма. В эксперименте производится настройка $\mathbf{\Gamma}$ по частям, то есть алгоритм запускается отдельно для каждого слоя. Результатом работы является вектор, нулевые элементы которого соответствуют нейронам, которые исключаются из структуры.

Функция ошибки и критерии качества модели

Для оптимизации структуры предлагается использовать композитную функцию ошибки. Она состоит из двух слагаемых. Первое слагаемое соответствует точности восстановления зависимой переменной. Второе слагаемое это точность реконструкции независимой переменной автокодировщиком. Задача (14) является задачей минимизации функции S. Она включает слагаемые (16) и (19) для оптимизации параметров модели (2)

$$f = \underbrace{\sigma_k \circ \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}}\boldsymbol{\sigma}_{k-1} \circ \underbrace{\mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \circ \underset{n_2\times 1}{\mathbf{W}_2}\boldsymbol{\sigma}_1 \circ \underset{n_1\times n}{\mathbf{W}_1}\underset{n\times 1}{\mathbf{x}}}_{E_{\mathbf{x}}}}_{S}. \tag{15}$$

Первое слагаемое $E_{\mathbf{x}}$ — это функция ошибки реконструкции объекта стеком автокодировщиков. Второе слагаемое S — это функция ошибки нейросети.

При выборе моделей используется три вида критериев качества: точность, устойчивость и сложность.

Точность. При решении задачи регрессии, то функция ошибки имеет вид:

$$S = \sum_{i\in\mathcal{I}} \big(y_i - f(\mathbf{x}_i)\big)^2. \tag{16}$$

Эта функция ошибки включает в себя полученные предсказания модели и значения зависимых переменных. В задачах прогнозирования точность аппроксимации имеет вид:

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|. \tag{17}$$

При включении в модель (2) метода главных компонент или автокодировщика, метки объектов не используются. Функция ошибки штрафует невязки восстановленного объекта:

$$E_{\mathbf{x}} = \sum_{i\in\mathcal{I}} \|\mathbf{x}_i - \mathbf{r}(\mathbf{x}_i)\|_2^2, \tag{18}$$

где $\mathbf{r}(\mathbf{x})$ это линейная реконструкция объекта $\mathbf{x}$. Функция (18) с аддитивной регуляризацией:

$$E_{\mathbf{x}} = \frac{1}{2m}\sum_{i=1}^{m} \|\mathbf{r}(\mathbf{x}_i, \mathbf{W}_{\mathrm{AE}}) - \mathbf{x}_i\|^2 + \lambda^2 \|\mathbf{W}\|^2_{\mathrm{Frobenius}}, \tag{19}$$

где m — число элементов в обучающей выборке. Параметры автокодировщика

$$\mathbf{W}_{\text{AE}} = \{\mathbf{W}', \mathbf{W}, \mathbf{b}', \mathbf{b}\} \tag{20}$$

оптимизированы таким образом (18), чтобы приблизить реконструкцию $\mathbf{r}(\mathbf{x})$ к исходному вектору $\mathbf{x}$.

Процедура оптимизации параметров композитной функции (15): 1) оптимизируются параметры модели согласно (19), 2) заданные параметры фиксируются, 3) оптимизируются параметры согласно (16).

Сложность. Введем отношение порядка $\succ$ на множестве значений сложности. Это отношение задается множеством параметров модели:

1) один параметр: $w \in \mathbb{R}^1 \succ w \in \lambda_1[0,1] + \lambda_0 \succ w \in c + \lambda_0$,

2) вектор (нейрон): $\mathbf{w} \in \mathbb{R}^n \succ \|\mathbf{w}\|^2 = 1 \succ \mathbf{w} = \text{const}$,

3) матрица (слой): $\mathbf{W} \in \mathbb{R}^{c \times n} \succ \mathbf{W}^\mathsf{T}\mathbf{W} = \mathbf{I} \succ \mathbf{W} = \text{const}$.

Следующие модели упорядочены по возрастанию сложности:

1) линейная регрессия, $\sigma' = \text{id}, \sigma = \text{id}, \mathbf{W} = \mathbf{I}_n$,

2) линейная регрессия и метод главных компонент, $\sigma' = \text{id}, \mathbf{W}^\mathsf{T}\mathbf{W} = \mathbf{I}_n$,

3) линейная модель и автокодировщик, $\mathbf{W}^\mathsf{T}\mathbf{W} \neq \mathbf{I}_n$,

4) линейная модель и стэк автокодировщиков, представимый в виде суперпозиции (12),

5) двухслойная нейронная сеть,

6) глубокая нейронная сеть.

Устойчивость — это минимум дисперсии функции ошибки (16):

$$\mathsf{D}(S) \to \min . \tag{21}$$

При вычислении устойчивости выборка считается фиксированной и изменение устойчвости считается зависящим только от структуры и параметров модели.

Вычислительный экперимент

Исследуется процедура оптимизации структуры нейросети с сохранением качества аппроксимации. Структура оптимизируется с помощью генетического алгоритма. Цель вычислительного эксперимента состоит в определении оптимальной позиции разделения автокодировщиков и нейронной сети, а так же исследовании зависимости точности, сложности и устойчивости модели от способа задания структуры. Исходный код находится на Github [20].

Процедура построения модели включает в себя следующие шаги:

1. Задание суперпозиции. Для оптимизации параметров используется метод стохастического градиентного спуска.

2. Структура модели оптимизируется генетическим алгоритмом.

3. Для сравнения сложности структуры и исследования зависимости ошибки от сложности вводится отношение порядка на $\mathbf{\Gamma}$.

Структура вычислительного эксперимента представлена на рис. 1.

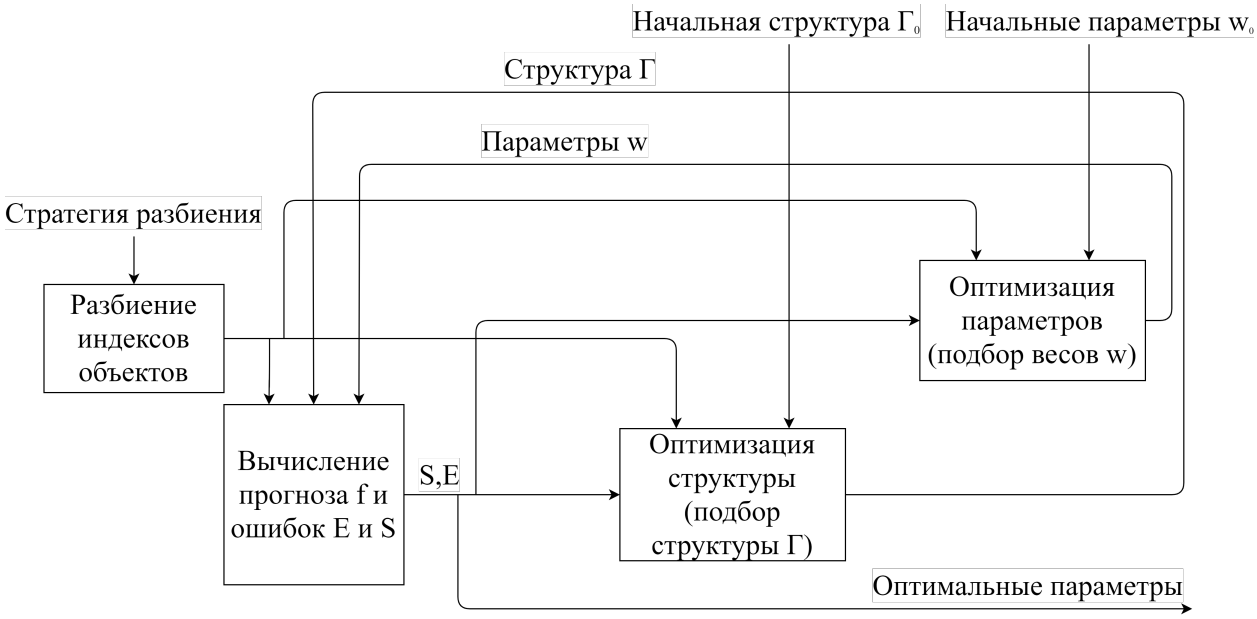


Рис. 1: Функциональная схема программной системы для оптимизации структуры суперпозиции

Таблица 1: Описание выборок для экспериментов

Выборка $\mathfrak{D}$	Размер train	Размер val	Размер test	Объекты	Признаки
Credit Card	18000	6000	6000	30000	35
Protein	27438	9146	9146	45730	9
Airbnb	6298	2100	2100	10498	16
Wine quality	2938	980	980	4898	11
Synthetic	1200	400	400	2000	30

Описание процедуры оптимизации.

1. Инициализируются начальные параметры и структура.

2. Используется метод бустрепа для генерации дополнительных выборок. Бустреп возвращает набор выбранных индексов объектов сообразно выбранной стратегии, которая преследует следующие цели:

   a) обучение: оптимизация параметров $\mathbf{w}$,

   b) контроль $\mathbf{\Gamma}$,

   c) валидация полученной модели f,

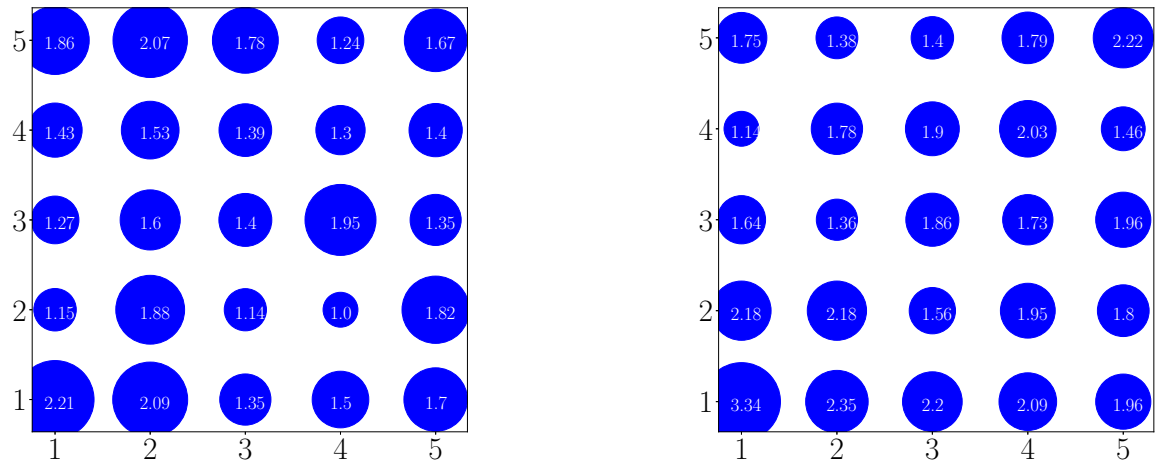
   d) вычисление дисперсии функции ошибки S.

3. Вычисление ошибки предсказания S и ошибки реконструкции $E_{\mathbf{x}}$.

4. Оптимизация параметров алгоритмом стохастического градиентного спуска.

5. Оптимизация структуры генетическим алгоритмом.

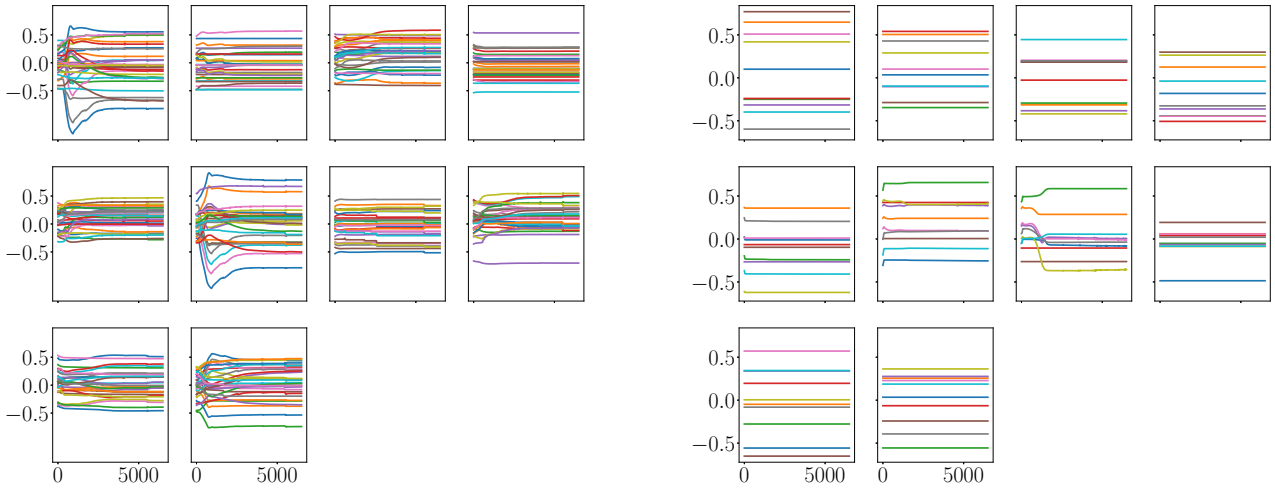
6. Итеративное повторение шагов 3–5.



(а) (б)

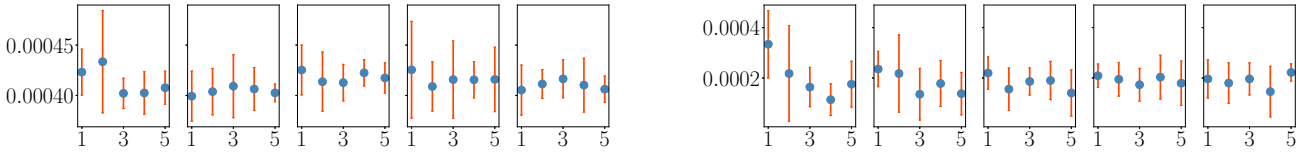
Рис. 2: Ошибка (16) в зависимости от конфигурации модели *а*) первый подход, *б*) второй подход. По горизонтали отложено количество слоев в автокодировщике, по вертикали — в полносвязной сети.

Наборы данных. Качество предложенного подхода к построению модели оценивается на нескольких реальных наборах данных и одном синтетическом наборе. Выборки взяты из открытого репозитория данных для машинного обучения [7]. Описание всех выборок представлено в табл 1. Синтетический набор данных состоит из признаков с



(а) (б)

Рис. 3: Параметры нейронов в зависимости от числа итераций: а) автокодировщик, б) полносвязная сеть. Каждый график соответствует нейрону по порядку справа налево и сверху вниз.



(а) (б)

Рис. 4: Ошибка и ее дисперсия в зависимости от структуры модели а) первый подход, б) второй подход. Порядковый номер графика и точки на графике соответствуют разному количеству слоев соответствующей модели.

Таблица 2: Результат применения генетического алгоритма для прореживания сети

Выборка $\mathfrak{D}$	Ошибка сети с прореживанием	Ошибка сети без прореживания	Сложность без прореживания	Сложность после прореживания
Credit Card	0.3204 ± 0.0032	0.2681 ± 0.0034	68	25
Protein	4.4968 ± 0.0238	4.4968 ± 0.0238	16	1
Airbnb	35.0773 ± 0.5909	33.9163 ± 0.5978	32	12
Wine quality	0.5818 ± 0.0147	0.5941 ± 0.0149	20	4
Synthetic,10^{-3}	0.3005 ± 0.0081	0.303 ± 0.0079	60	12

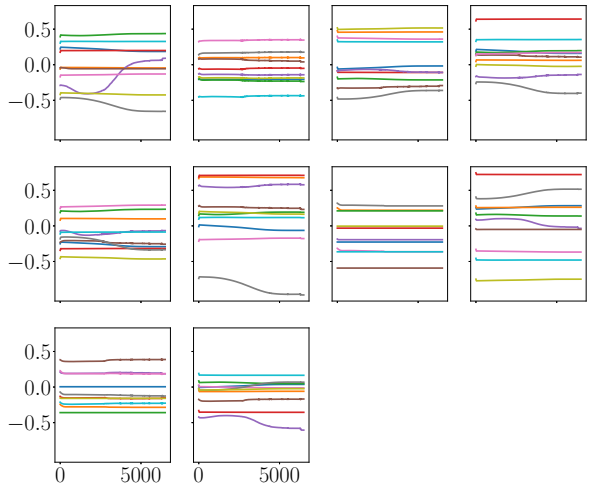


Рис. 5: Параметры нейронов в зависимости от числа итераций, полносвязная сеть, второй подход. Каждый график соответствует нейрону по порядку справа налево и сверху вниз.

различными свойствами ортогональности и коррелированности друг с другом и к целевой переменной. Процедура генерации синтетических данных описана в работе [2]. Возможны следующие конфигурации синтетических данных.

1. Неполный и скоррелированный: набор данных, содержащий коррелирующие признаки, ортогональные с целевому вектору.

2. Адекватный и случайный: набор данных, содержащий случайные признаки, и имеющий один признак, апроксимирующий целевой вектор.

3. Адекватный и избыточный: набор данных, содержащий признаки, коррелирующие с целевым вектором.

4. Адекватный и скорреллированный: набор данных, содержащий ортогональные признаки, и признаки, коррелирующие с ортогональными. Целевой вектор является суммой ортогональных векторов.

Каждый набор данных разбивается на три части.

1. Обучающая выборка — 60% от исходного набора. На этой выборке модель тренируется, и фиксируются значения параметров.

2. Валидационная выборка — 20% от исходного набора. На этой выборке применяется генетический алгоритм, который ищет оптимальную структуру.

3. Тестовая выборка — 20% от исходного набора. Она никак не участвует в оптимизации структуры модели. Эта выборка используется только для контроля качества — сравнение модели исходной и оптимизированной структуры, а так же сравнение с другими алгоритмами прореживания сетей.

Решается задача восстановления регрессии, то есть зависимой переменной является $y \in \mathbb{R}$. В процессе работы были рассмотрены два подхода решения задачи, и соответственно две структуры нейронной сети.

Первый подход. Автокодировщик преобразует входные векторы $\mathbf{x}$, которые затем подаются на вход полносвязной нейронной сети.

Второй подход. Оптимизируются параметры автокодировщика, его параметры фиксируются и предпоследний слой соединяется с полносвязной нейросетью, оптимизирутся параметры модели. Предпоследний слой содержит меньшее количество параметров по сравнению с размерностью входного пространства признаков, определяемое количеством нейронов в этом слое. То есть полносвязная сеть получает на вход скрытое представление исходной независимой переменной.

Параметры в обоих сетях инициализированы нормальным распределением с нулевым средним и смещением $\sqrt{\frac{2}{N_{\text{in}}+N_{\text{out}}}}$, где N_{in} — число входных признаков,
а N_{out} — число признаков на выходе слоя. Такая инициализация параметров была предложена в [5]. Она выбрана экспериментальным путем, как показывающая наилучший результат точности аппроксимации. Каждая из сетей обучалась в течение 500 итераций обновления параметров и размер пакета обучения равен 128. В качестве функции активации в слоях автокодировщика используется $\text{Relu}(x) = \max(0, x)$, для полносвязной сети тоже Relu, но в последнем слое id.

Для вычисления ошибки (16) используются выходные значения полносвязной сети. Варьируется число промежуточных слоев автокодироващика и полносвязной сети от одного до пяти. Рассмотривается декартово произведение двух
множеств: $[1, 2, 3, 4, 5] \times [1, 2, 3, 4, 5]$. Число нейронов в каждом скрытом слое одинаково для любой сети и равно десяти. Для каждой конфигурации считается ошибка (16). Качество оценивается на синтетическом наборе данных. Полученный результат представлен на рис. 2. Размер пузыря пропорционален полученной ошибке. Каждая конфигурация сети обучалась на наборе, полученном с помощью бустреп-метода из данных, взятых для обучения. Количество итераций процедуры бустреп-метода равно 10. Ошибка считалась на отложенном наборе данных для тестирования. Видно, что при увеличении числа слоев полносвязной сети ошибка в основном падает. Минимальная ошибка достигается при конфигурации: четыре слоя автокодировщика и два слоя полносвязной сети для первого подхода, и один слой автокодировщика и четыре слоя полносвязной сети для второго подхода. Данную конфигурацию возьмем для дальнейшего исследования параметров модели, соответственно используя второй подход. На рис. 3 показано изменение значений параметров в первом скрытом слое автокодировщика и полносвязной сети в зависимости от числа итераций. Видно, что многие параметры, и даже нейроны не изменяют свое начальное состояние. Вообще говоря, такие параметры модели не влияют на ошибку и могут быть свободно удалены без потери качества.

На рис. 4 представлена дисперсия (21) и значение ошибки (16) в зависимости от числа слоев в автокодировщике и полносвязной сети. Дисперсия была получена с помощью десяти итераций бустрапирования обучающей выборки для каждой конфигурации. С увеличением числа слоев в автокодировщике при использовании первого подхода снижаются ошибка и дисперсия ошибки.

С помощью описанных двух подходов получается оптимальная архитектура соединения автокодировщика и полносвязной сети. Далее применяется описанный ранее генетический алгоритм для прореживания сети и уменьшения ее сложности. Алгоритм применяется на нейронах каждого слоя сети. В табл. 2 приведены результаты применения генетического алгоритма для исследуемых наборов данных, качество алгоритма оценивается по тестовой выборке. В качестве ошибки выступает (17), а в качестве сложности алгоритма выступает число ненулевых нейронов. Под нулевым нейроном понимается нейрон, все параметры которого равны нулю.

Заключение

В представленной работе исследовано два подхода построения модели, состоящей из автокодировщика и нейронной сети и имеющей композитную функцию ошибки. Представлены подходы по поиску оптимальной точки разделения автокодировщика и нейронной сеть. Исследовано применение генетического алгоритма для оптимизации структуры и снижения сложности. Работа предложенного алгоритма исследовалась на пяти различных наборах данных. Из приведенной таблицы можно сделать вывод, что предложенный алгоритм работает и позволяет существенно снизить сложность модели без потери качества аппроксимации.

Для дальнейшего развития предложенной идеи предполагается построить дифференцируемую функцию ошибки, которая будет зависеть от сложности модели. При построении такой функции возможно применять различные оптимизационные алгоритмы для уменьшения ошибки и одновременно сложности модели. Предполагается ввести новый иерархический способ описания структуры модели.

Список литературы

[1] *Бахтеев О. Ю., Стрижов В. В.* Выбор моделей глубокого обучения субоптимальной сложности // Автомат. и телемех., 2018. Вып. 8. С. 129–147.

[2] *Katrusa A. M., Strijov V. V.* Stress test procedure for feature selection algorithms // Chemometrics and Intelligent Laboratory Systems, 2015. Vol. 142. P. 172–183.

[3] *LeCun Y., Denker J. S., Solla S., Howard R. E., Jackel L. D.* Optimal brain damage // In Advances in Neural Information Processing Systems 2, NIPS. — Denver, Colorado, USA: 1989. P. 598–605.

[4] *Pao C. P.* Линейные статистические методы и их применение / пер. с англ. – М.: Наука, 1968. С. 548. (*Radhakrishna Rao C.* Linear statistical inference and its application. – 1nd ed. – New York: Wiley, 1968. P. 548.)

[5] *Glorot X., Bengio Y.* Understanding the difficulty of training deep feedforward neural networks // 13th International Conference on Artificial Intelligence and Statistics Proceedings, AISTATS. — Sardinia, Italy: 2010. P. 249–256.

[6] *Cybenko G. V.* Approximation by Superpositions of a Sigmoidal function // Mathematics of Control Signals and Systems, 1989. Vol. 2(4) P. 303–314.

[7] UCI Machine Learning Repository. 2007. https://archive.ics.uci.edu/ml

[8] *Dong X., Chen S., Pan S.* Learning to Prune Deep Neural Networks via Layer-wise Optimal Brain Surgeon // Advances in Neural Information Processing Systems 30, NIPS. — Long Beach, California, USA: 2017. P. 4857–4867.

[9] *Hassibi B., Stork D. G.* Second order derivatives for network pruning: Optimal brain surgeon // In Advances in Neural Information Processing Systems 6, NIPS. — Denver, Colorado, USA: 1993. P. 164—171.

[10] *Molchanov P., Tyree S., Karras T., Aila T., Kautz J.*. Pruning convolutional neural networks for resource efficient transfer learning. 2016. https://arxiv.org/pdf/1611.06440.pdf

[11] *Chaber P., Ławryńczuk M.* Pruning of recurrent neural models: an optimal brain damage approach // Nonlinear Dynamics, 2018. Vol. 92(2). P. 763.

[12] *Han S., Mao H., Dally W. J.* Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. 2015. https://arxiv.org/pdf/1510.00149.pdf

[13] *Kingma D. P., Salimans T., Welling M.* Variational Dropout and the Local Reparameterization Trick // Advances in Neural Information Processing Systems 28 / Ed. by C. Cortes, N. D. Lawrence, D. D. Lee et al. — Curran Associates, Inc., 2015. — Pp. 2575—2583.

[14] *Molchanov D., Ashukha A., Vetrov D.* Variational Dropout Sparsifies Deep Neural Networks // 34th International Conference on Machine Learning Proceedings, ICML. — Sydney, NSW, Australia: 2017. Pp. 2498—2507.

[15] *Arabasadi Z., Alizadehsani R., Roshanzamir M., Moosaei H., Yarifard A. A.* Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm // Computer Methods and Programs in Biomedicine, 2017. Vol. 141. P. 19–26.

[16] *Hinton G. E., Salakhutdinov R. R.* Reducing the dimensionality of data with neural networks // Science, 2006. Т. 313. №. 5786. С. 504–507.

[17] *Srivastava N. et al.* Dropout: a simple way to prevent neural networks from overfitting // The journal of machine learning research, 2014. Т. 15. №. 1. С. 1929–1958.

[18] *Elsken T., Metzen J. H., Hutter F.* Neural architecture search: A survey // arXiv preprint arXiv:1808.05377, 2018.

[19] *Hutter F., Kotthoff L., Vanschoren J.* Automated Machine Learning-Methods, Systems, Challenges // Automated Machine Learning, 2019.

[20] *Потанин М. С., Вайсер К. О., Жолобов В. А., Стрижов В. В.* Приложение к статье: вычислительный эксперимент по выбору универсальной модели и базовые теоремы, 2019. https://github.com/MarkPotanin/GeneticOpt